# SOC-GA 2332 Intro to Stats Lab 5

Wenhao Jiang

10/04/2024

## Logistics & Announcement

**Problem Set 2** is due 10/25

First, load packages to your environment. Today we will use several new packages. Please install them before you run the following chunks.

```
knitr::opts_chunk$set(echo = TRUE)

library(tidyverse)
library(foreach)
library(stargazer)
library(ggcorrplot)
library(psych)
```

## Part 1: Data Simulation and How to Simulate Regressions

Data simulation is the opposite of data analysis. You create a data set through simulation and then "understand" it using the analytically tools you learned.

### Advantages for doing data simulation

1. *The truth is known*: We know the values of simulated parameters, and we can therefore compare them with the model estimates. This gives us a better understanding of how well the model performs.
2. *Understanding parameters through tuning*: Sometimes the effect of one or more parameters is unclear. Being able to tune the parameters and observe how they affect the resultant data helps us better understand the parameters.
3. *Evaluating the bias and efficiency of statistics*: As we have practiced in PS1, by simulating a virtual population, we can directly observe how the sampling procedure affects the variability of the sample statistics. In general, through simulation, we can generate sampling distribution of any statistics we are interested in, and evaluate their bias and efficiency.
4. *Understanding models*: Finally, data simulation provides proof that you understand a model: If you can simulate data under a certain model, then it is likely that you really understand that model.

### Simulation from a stochastic process

For a social system, we always take into account the randomness in social life when we simulate data. In other words, the data is generated from a **stochastic process** in which the state of the system cannot be precisely predicted given its current state and even with a full knowledge of all the factors affecting that process.

In concrete, it means that we always include an error/residual term when simulating a social process. This error/residual term conveys the unpredictability and randomness of social lives.

### Simulate a bivariate relationship

For example, let's simulate a **population** data in which the *years of education* affect one's *income rank* according to the following equation:

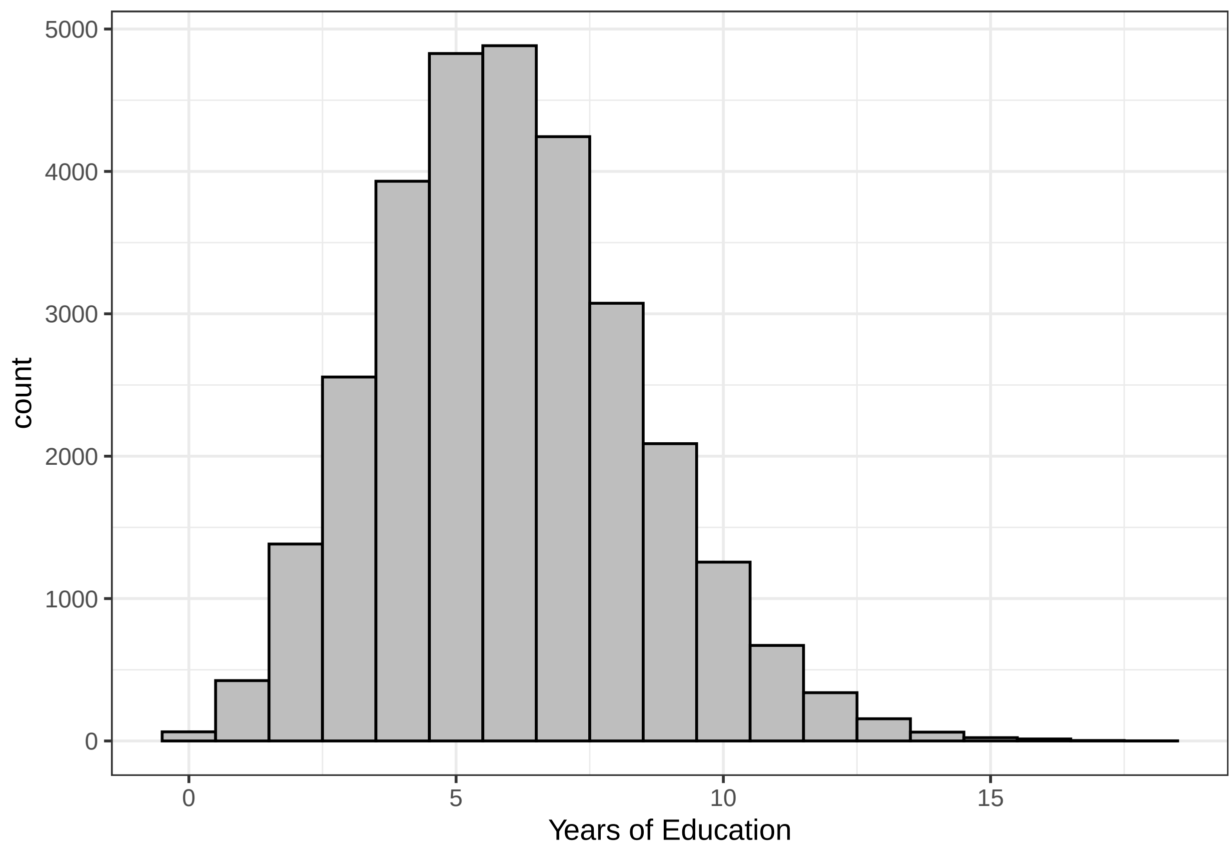
$$I_i = 10 + 6 \cdot E_i + \epsilon_i$$

We will first simulate *years of education* - the independent variable (IV), then *income rank* - the dependent variable (DV) according to the above equation.

We simulate years of education using `rpois()`, a function that generates a random Poisson distribution with a specified parameter $\lambda$. You can learn more about the distribution here.

```
## simulate IV (edu level)
set.seed(1234)
edu <- rpois(30000, lambda = 6) ## rpois: Random Poisson Distribution with parameter lamda

## lot histogram of IV
edu %>%
  as_tibble() %>%
  ggplot(aes(value)) +
  geom_histogram(color = "black", fill = "grey", binwidth = 1) +
  labs(x = "Years of Education") +
  theme_bw()
```



```r
## summary statistics
summary(edu)
```

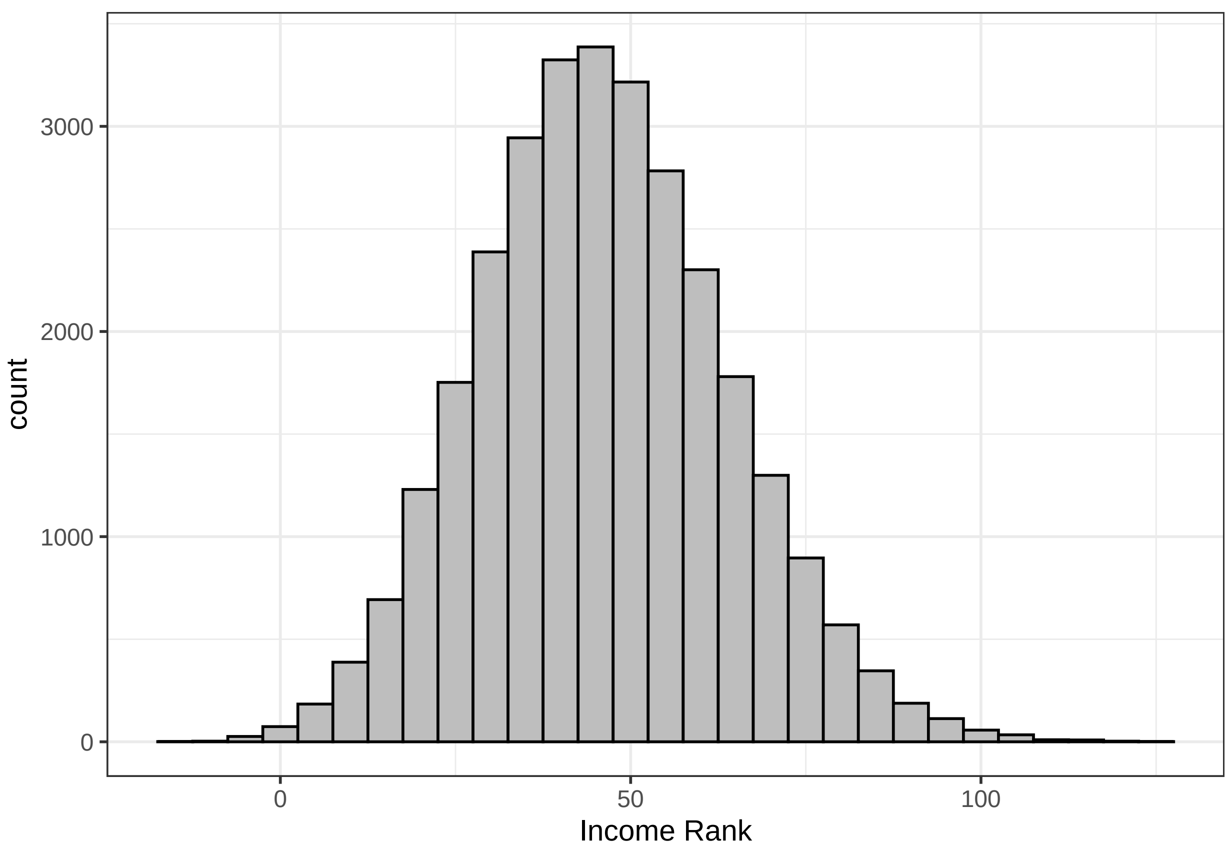
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   4.000   6.000   6.022   8.000  18.000
```

When simulating earning, since we are simulating a stochastic process, we always add an error term, noted as $\epsilon_i$ in the above equation. Note that normally this error term is modeled using `rnorm()` assuming **the error term is normally distributed with a mean of 0 and a sd that is a constant value**, so that the errors vary following the same random pattern across all values of the IV.

But we might need to change this when we want to simulate data that **violate the homoskedasticity assumption**, which means the error term is not purely random but dependent on the value of the IV.

```r
## simulate DV
set.seed(1234)
earn <- 10 + 6*edu + rnorm(30000, 0, 10) ## add a random error using rnorm()

## plot histogram of DV
earn %>%
  as_tibble() %>%
  ggplot(aes(value)) +
  geom_histogram(color = "black", fill = "grey", binwidth = 5) +
  labs(x = "Income Rank") +
  theme_bw()
```




```r
## summary statistics
summary(earn)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -13.27   33.81   45.42   46.14   57.73  122.96
```

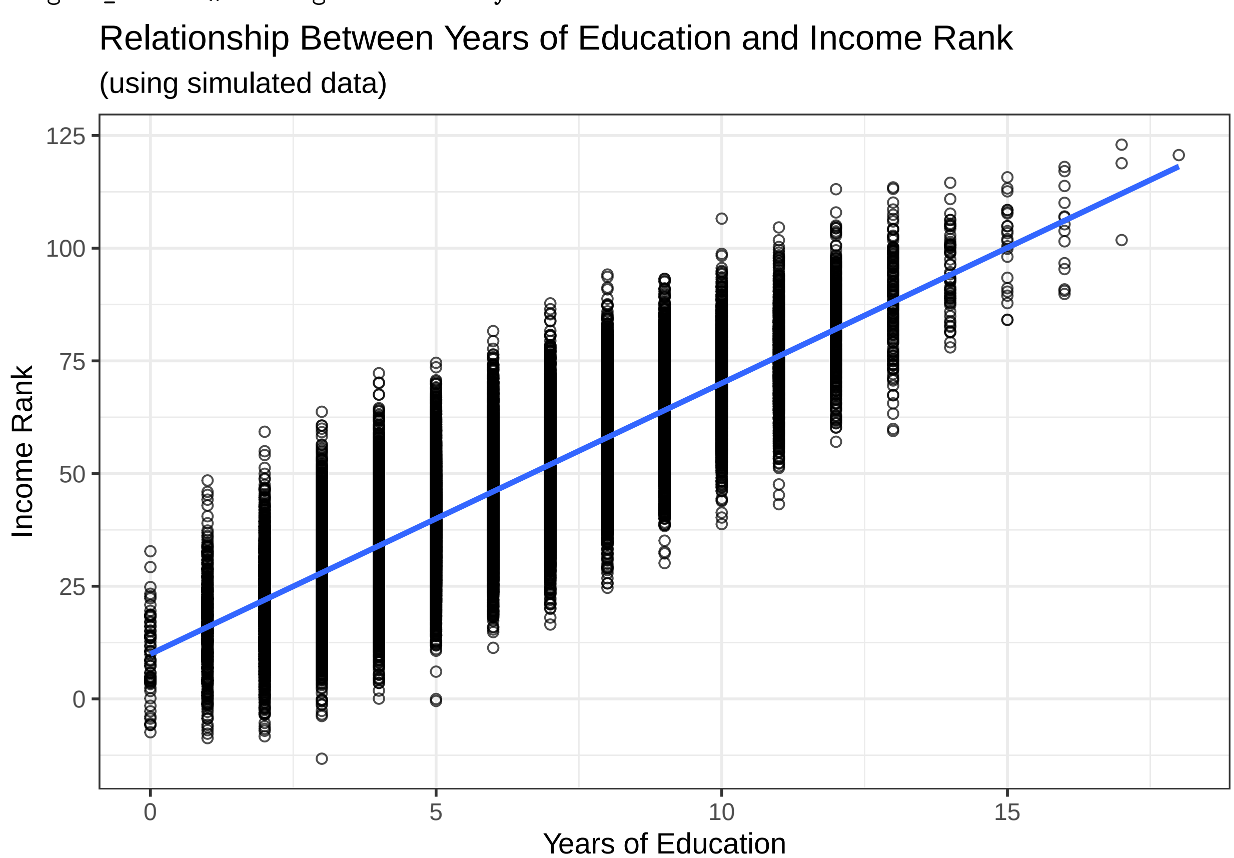
```r
## combine data frame
df <- data.frame(x_edu = edu,
              y_earn = earn)

## summary statistics of the data frame
describe(df)
```

```
##        vars     n  mean    sd median trimmed   mad    min    max  range skew
## x_edu     1 30000  6.02  2.44   6.00    5.93  2.97   0.00  18.00  18.00 0.40
## y_earn    2 30000 46.14 17.73  45.42   45.77 17.66 -13.27 122.96 136.23 0.22
##        kurtosis   se
## x_edu      0.16 0.01
## y_earn     0.05 0.10
```

```r
## scatter plot with a fitted lm line
df %>%
  ggplot(aes(x = x_edu, y = y_earn)) +
  geom_point(shape = 1, alpha = 0.7) +
  geom_smooth(method = "lm") +
  labs(title = "Relationship Between Years of Education and Income Rank",
       subtitle = "(using simulated data)",
       x = "Years of Education",
       y = "Income Rank") +
  theme_bw()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



## Fit OLS to sampled data

We can fit a regression model to the sampled data from the simulated population, and compare the result with the "true" relationship. As you can see our modeling result is quite close to the "true" parameter values.

```r
## sample 300 obs
sample <- df[sample(1:dim(df)[1], 300), ]

## run a model
m_simu <- lm(y_earn ~ x_edu, data = sample)

summary(m_simu)
```

```
##
## Call:
## lm(formula = y_earn ~ x_edu, data = sample)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26.2909  -6.2295   0.2123   6.8204  25.3658
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.394      1.522   6.171 2.21e-09 ***
## x_edu          6.058      0.237  25.555  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.41 on 298 degrees of freedom
## Multiple R-squared:  0.6867, Adjusted R-squared:  0.6856
## F-statistic: 653.1 on 1 and 298 DF,  p-value: < 2.2e-16
```

## `stargazer` for regression tables

In displaying regression tables, a commonly used package is `stargazer`, its main function `stargazer()` can format modeling results to neat regression tables. You can set the output type `type = "latex` and copy the output to Overleaf or other LaTeX editors, the code will automatically render to a cleanly formatted regression table.

```r
## use stargazer to display regression tables
## for quick view in R:
stargazer(m_simu, type = "text")
```

```
##
## ===============================================
##                         Dependent variable:
##                     ---------------------------
##                               y_earn
## -----------------------------------------------
## x_edu                        6.058***
##                               (0.237)
##
## Constant                     9.394***
##                               (1.522)
##
## -----------------------------------------------
```

```
## Observations                      300
## R2                               0.687
## Adjusted R2                      0.686
## Residual Std. Error      10.411 (df = 298)
## F Statistic          653.067*** (df = 1; 298)
## ================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```

```
## for PDF:
## stargazer(m_simu, type = "latex")
```

A nice feature of `stargazer` is that it allows you to customize what statistics to add or omit in your table. For example, if we want to omit Residual Std. Error and the F Statistic:

```
## omit some statistics
stargazer(m_simu, m_simu, type="latex", omit.stat=c("ser", "f"),
          dep.var.labels = "Earning",
          covariate.labels = c("Education","Constant"))
```

Take a look at p.22 on stargazer's documentation. You will find all statistics abbreviations you can use for omission.

You can also change the dependent variable label, independent variable labels, column labels, etc. Try Google them when you want to customize.

## Part 1 Exercise:

1. Simulate the population data with the same relationship as specified above, only **changing the residual term from having a sd = 10 to sd = 50**. Follow the steps demonstrated above, create a sampled dataframe and fit an OLS model to your sampled data. Create a scatterplot of your data. *Note:* Remember to `set.seed()`!

```
## your simulation
```

2. Then, using the formula we have learned in the lecture, calculate the following statistics in `R`:

(a) The value of TSS
(b) $R^2$
(c) Verify that $R^2 = \rho^2$
(d) $se_{\beta_1}$
(e) Construct the 95% confidence interval for $\beta_1$ (You can use the $\beta_1$ value reported in your OLS modeling result)

```
## TSS

## SSE

## R^2

## pho^2

## SE of beta1

## 95% CI
```

- The **sum of square error (SSE)** for Y is the sum of square errors for the fitted OLS model:

$$SSE = \sum \epsilon^2 = \sum (y - \hat{y})^2$$

- The **total sum of squares (TSS)** for Y is the sum of square errors for the baseline OLS model (the "null model") that predict the value of Y without any X (the mean of Y, $\bar{Y}$, is used in the null model):

$$TSS = \sum (y - \bar{y})^2$$

- **R-squared** (coefficient of determination) is the proportional reduction in squared error when fitted with the OLS model in comparison with the null model:

$$R^2 = \frac{TSS - SSE}{TSS}$$

- The **mean square error (MSE)**:

$$MSE = \frac{SSE}{n-2}$$

- The **standard error of $\beta_1$ in bivariant OLS** regression:

$$se_{\beta_1} = \sqrt{\frac{MSE}{\sum (x - \bar{x})^2}}$$

3. Compare the scatterplot, the estimated $\beta_0$ and $\beta_1$, the $R^2$, and the $se_{\beta_1}$ in the demo OLS model output and your own simulated data model output, what do you observe? Why?

```
## our code here
set.seed(1234)
```

# Part 2: Multivariant Regression Using R

## Simulate Population

Suppose our outcome variable, $Y_i$, has the following data generating process in relation to education and income:

$$Y_i = 10 + 5 \cdot E_i + 3 \cdot I_i - 2 \cdot E_i \cdot I_i + \epsilon_i$$

Education is a random variable varies between 6 and 14 following a uniform distribution.

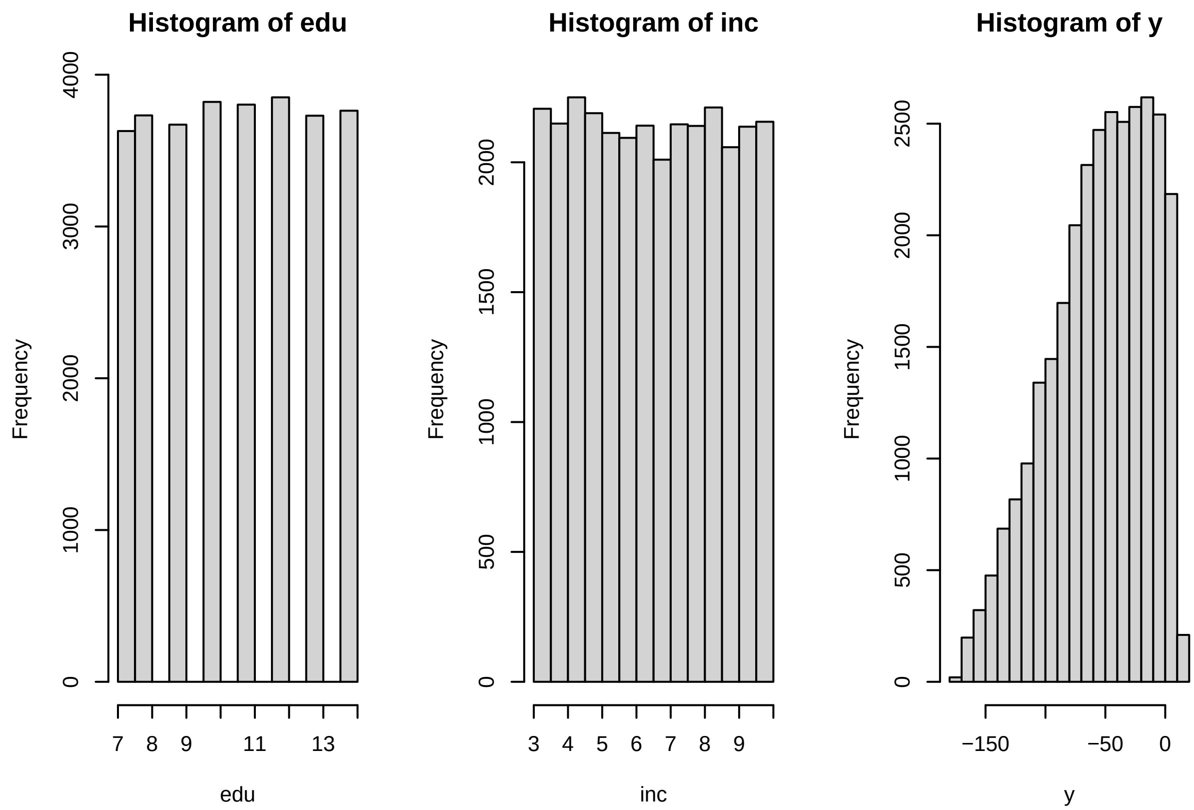
Income is a random variable varies between 3 and 10 following a uniform distribution.

The error term, $\epsilon$ follows a normal distribution with mean = 0 and sd = 2.

```
## simulate the data generating process for the population
set.seed(1234)
n <- 30000
edu <- runif(n, min = 6, max = 14) %>% ceiling() ## ceiling the value to get only integers
inc <- runif(n, min = 3, max = 10)
y <- 10 + 5*edu + 3*inc + (-2)*edu*inc + rnorm(n, mean = 0, sd = 2)

## combine variables into a data frame
df <- data.frame(edu = edu, inc = inc, y = y)

## check distribution
par(mfrow = c(1, 3)) ## arrange the following plots in 1 row and 3 cols
hist(edu)
hist(inc)
hist(y)
```




```r
## sample
set.seed(1234)
sample <- df[sample(1:dim(df)[1], 300), ]
```

### Multiple regression for a sample with no interaction term

First, let us estimate a multiple regression without interaction for a sample, and see how the OLS line fit in relation to the data. We also want to check how the model residuals behave in relation to fitted y.

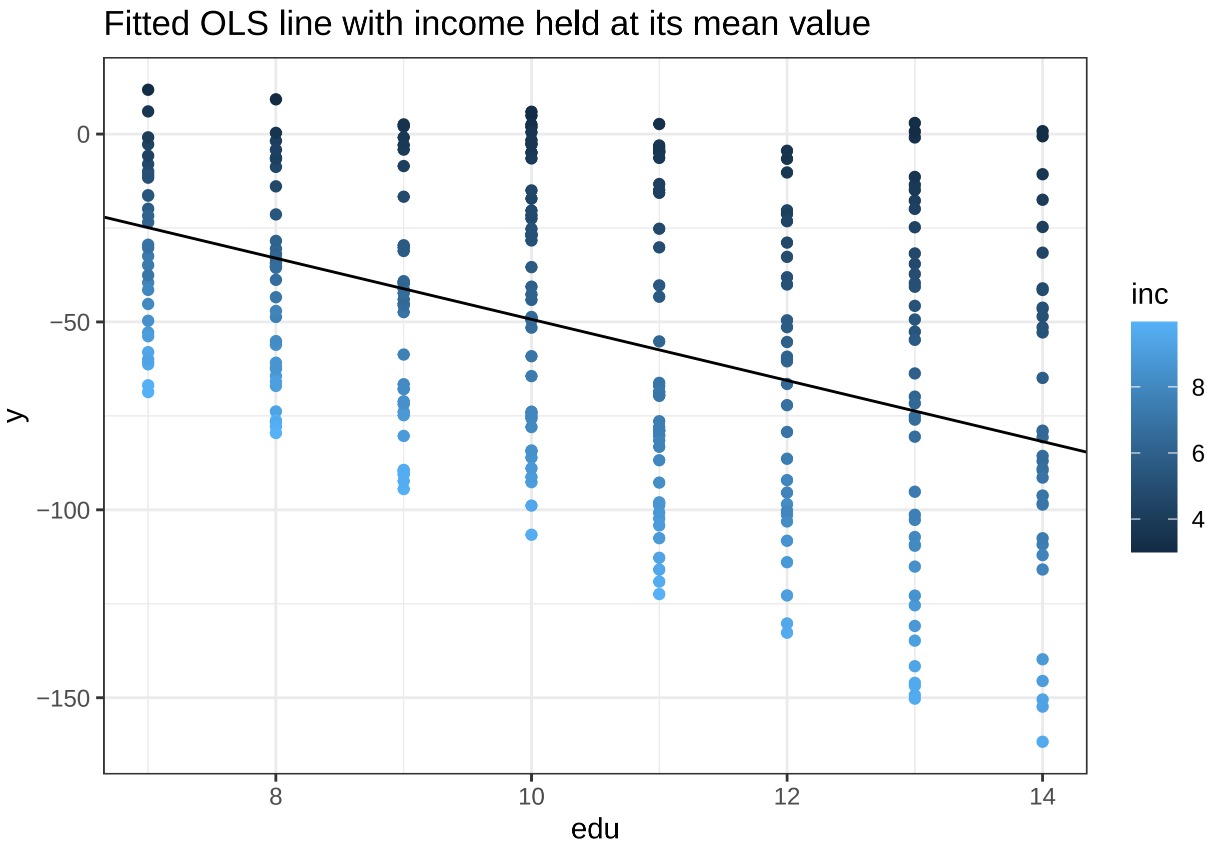
$$y_i = 10 + 3 \cdot I_i + 5 \cdot E_i + e_i$$

```r
## model without interaction
mod1 <- lm(y ~ edu + inc, data = sample)
summary(mod1)
```

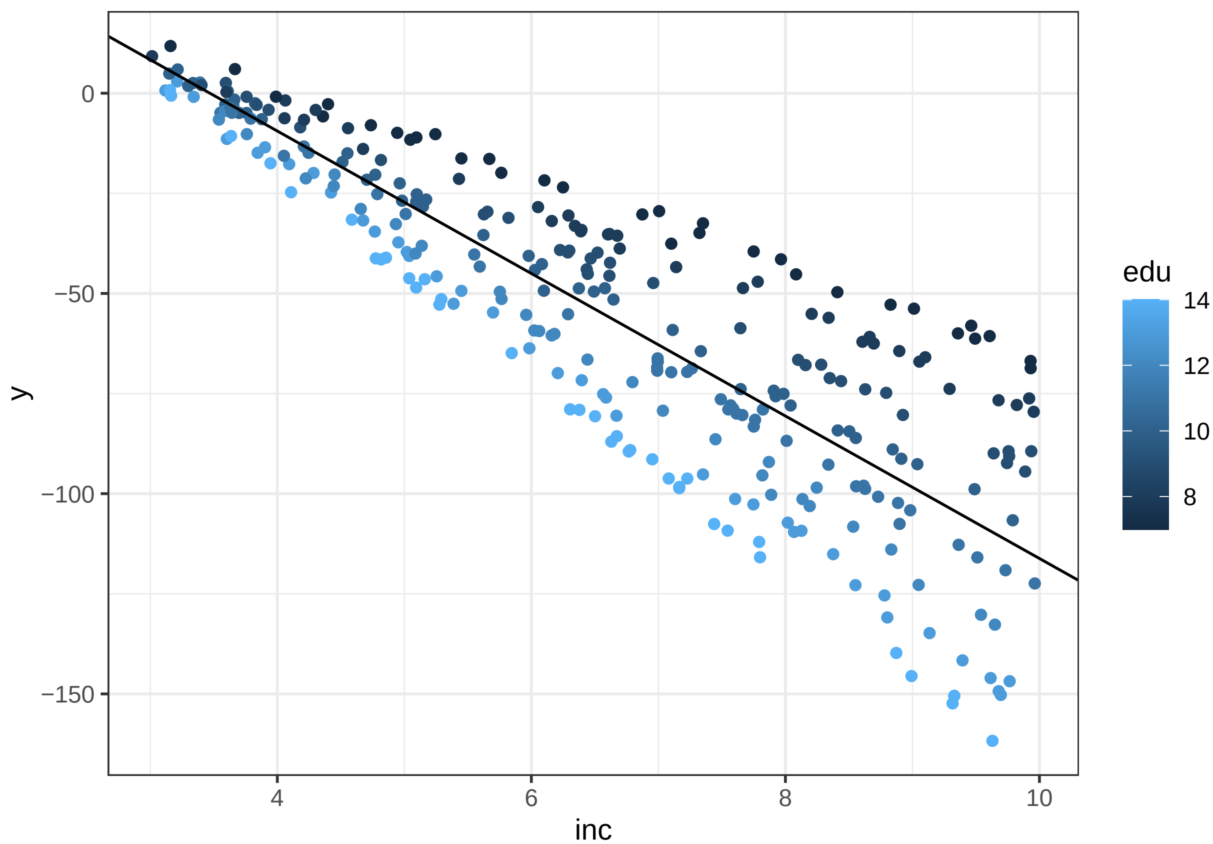
```
##
## Call:
## lm(formula = y ~ edu + inc, data = sample)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.8510  -4.2085  -0.6509   5.0628  23.4513
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 147.3015     3.1136   47.31   <2e-16 ***
## edu          -8.1323     0.2297  -35.41   <2e-16 ***
## inc         -17.7901     0.2570  -69.23   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.784 on 297 degrees of freedom
## Multiple R-squared:  0.9502, Adjusted R-squared:  0.9499
## F-statistic:  2835 on 2 and 297 DF,  p-value: < 2.2e-16
```

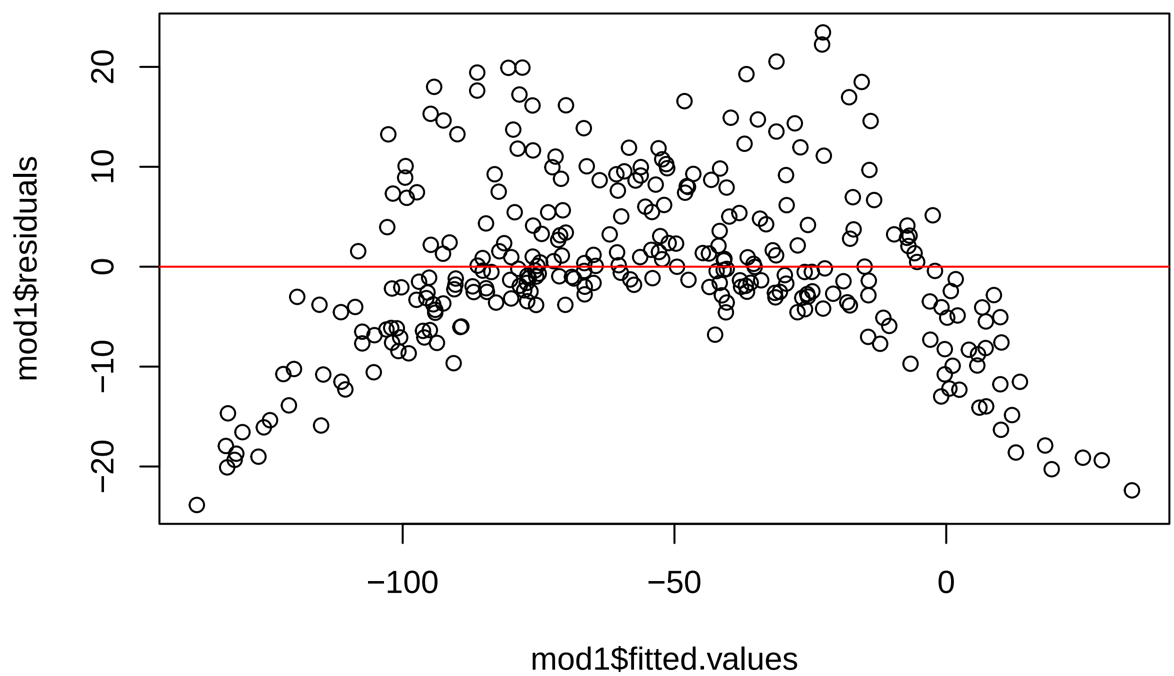
```
## a scatterplot of edu and y, with a fitted OLS line with income taking its mean value
sample %>%
  ggplot() +
  geom_point(aes(edu, y, color = inc)) +
  geom_abline(intercept = mod1$coefficients[1] + mod1$coefficients['inc']*mean(inc),
              slope = mod1$coefficients['edu']) +
  theme_bw() +
  labs(title = "Fitted OLS line with income held at its mean value")
```




```
## a scatterplot of inc and y, with a fitted OLS line with edu taking its mean value
sample %>%
  ggplot() +
  geom_point(aes(inc, y, color = edu)) +
  geom_abline(intercept = mod1$coefficients[1] + mod1$coefficients['edu']*mean(df$edu) ,
              slope = mod1$coefficients['inc']) +
  theme_bw()
```




```r
## check model residual behaviors
## residual vs. fitted y
plot(mod1$fitted.values, mod1$residuals)
abline(h = 0, col="red")
```




As we can see, when we did not taking into consideration of the interaction term, the model still captures the average effect of the predictor (while holding the other at constant). However, the scatterplot as well as the residual diagnostic also suggest that we are missing some important patterns.

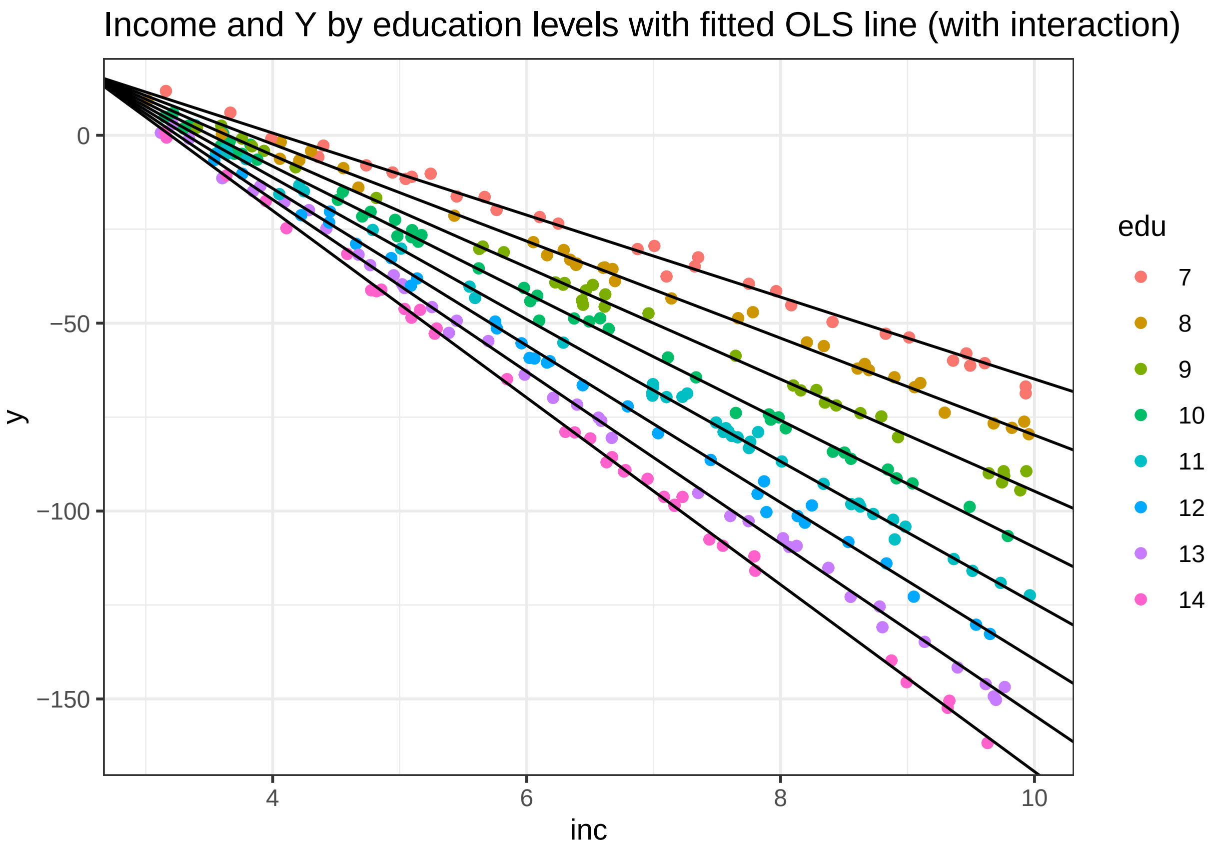
### Multiple regression with interaction:

Now we add the interaction term `edu*inc` into the model. We can plot the relationship between y and one of the x's and also plot the different slopes given the other x changes.

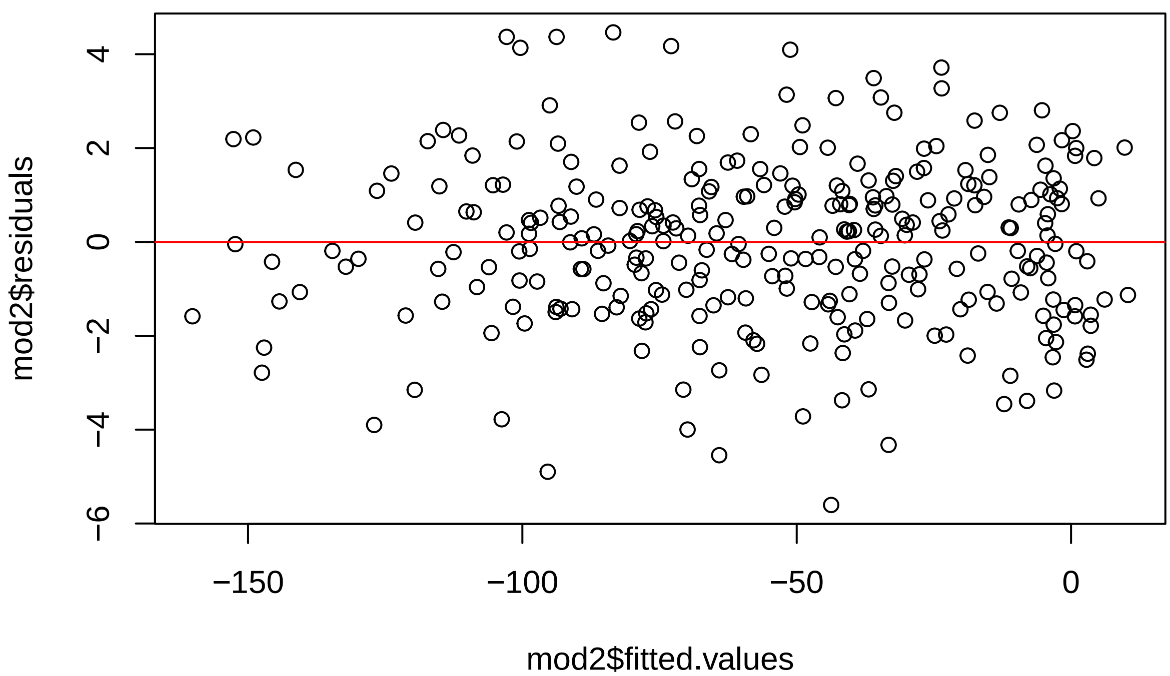
```r
## taking into account of the interaction:
mod2 <- lm(y ~ edu + inc + edu:inc, data = sample)
summary(mod2)
```

```
##
## Call:
## lm(formula = y ~ edu + inc + edu:inc, data = sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.6053 -1.2088  0.1302  1.0948  4.4638
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.03319    1.72818   5.227 3.26e-07 ***
## edu          5.03126    0.16032  31.383  < 2e-16 ***
## inc          3.05540    0.24874  12.284  < 2e-16 ***
## edu:inc     -1.99547    0.02331 -85.603  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.734 on 296 degrees of freedom
## Multiple R-squared:  0.9981, Adjusted R-squared:  0.998
## F-statistic: 5.095e+04 on 3 and 296 DF,  p-value: < 2.2e-16
```

```r
## relationship between inc and y when edu is changing:
sample %>%
  mutate(edu = as.factor(edu)) %>%
  ggplot(aes(inc, y, color = edu)) +
  geom_point() +
  geom_abline(intercept = mod2$coefficients[1] + mod2$coefficients['edu']*edu,
              slope = mod2$coefficients['inc'] + mod2$coefficients['edu:inc']*edu) +
  theme_bw() +
  labs(title = "Income and Y by education levels with fitted OLS line (with interaction)")
```




```r
## check model residual behaviors
## residual vs. fitted y
plot(mod2$fitted.values, mod2$residuals)
abline(h = 0, col="red")
```



### Interpret regression coefficient for model with interactions:

Interaction effects can be tricky to interpret. The safest way is to write down your models in equations and plug in values into the equations to figure out the difference.

For example, for Model 2, our regression equation is:

$$\hat{y}_i = 9.03 + 4.78 \cdot E_i + 2.51 \cdot I_i - 2.00 \cdot E_i \cdot I_i$$

Note that for the effect of `edu` and `inc`, it is no longer indicated by the coefficients of their individual terms. **Whenever you try to interpret the coefficient for variables that are included in an interaction term, you need to take into account the coefficient of the interaction term**.

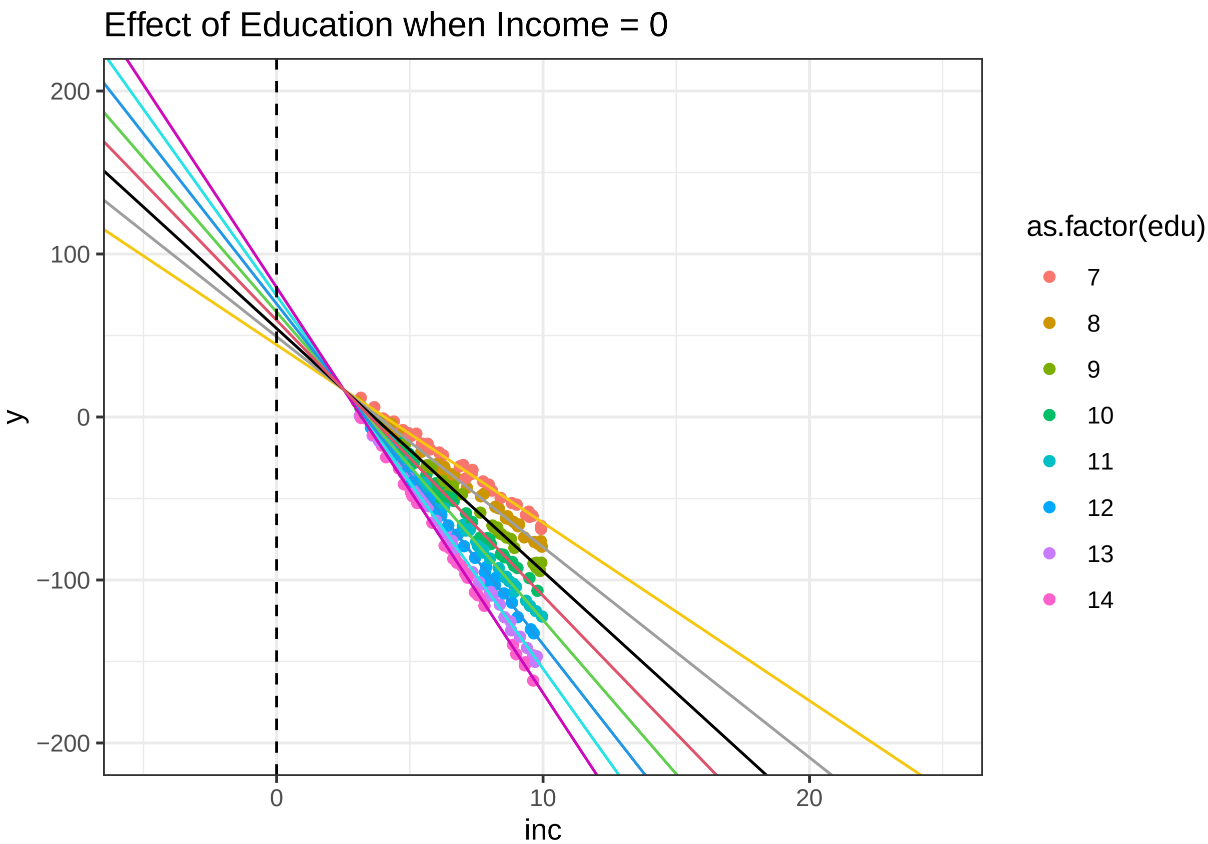
For example, let's examine the effect of `edu`. Holding all other variables constant, for **one unit increase in `edu`**, we have:

$$\Delta\hat{y} = 5.03 - 2.00 \cdot I$$

This means that **the effect of education depends on income**. And the coefficient in the term $5.03 \cdot E_i$ can only be interpreted as the effect of income **when income equals to 0**.

Let us look at the graphs on the effects of education on $y_i$ when there is an interaction effect between education and income:

```
## relationship between inc and y when edu is changing:
sample %>%
  ggplot(aes(inc, y, color = as.factor(edu))) +
  geom_point() +
  geom_abline(intercept = mod2$coefficients[1] + mod2$coefficients['edu']*edu,
              slope = mod2$coefficients['inc'] + mod2$coefficients['edu:inc']*edu,
              color = as.factor(edu)) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  xlim(-5, 25) +
  ylim(-200, 200) +
  theme_bw() +
  labs(title = "Effect of Education when Income = 0")
```




### Part 2 Exercise

- Import `lab5_earnings.csv` to your environment. Perform the following data cleaning steps: (1) If `age` takes the value 9999, recode it as `NA`; (2) Create a new variable `female` that equals 1 when `sex` takes the value `female`, and equals to 0 otherwise; (3) Create a new variable `black` that equals 1 when `race` is `black` and equals to 0 otherwise; (4) Create a new variable `other` that equals to 1 when `race` is 'other' and 0 otherwise.

- Use the `describe()` function from the `psych` package to generate a quick descriptive statistics of your data.

- Now, estimate the following models and display your model results in a single table using `stargazer(m_1, m_2, ..., m_n, type="text")`.

(1) Model 1: earn ~ age (baseline)
(2) Model 2: earn ~ age + edu
(3) Model 3: earn ~ age + edu + female
(4) Model 4: earn ~ age + edu + female + race
(5) Model 5: earn ~ age + edu + female + race + edu*female

- In Model 5, holding other variables constant, what will be the predicted difference in estimated mean earnings for a white man and a white women?

- In Model 5, holding other variables constant, what will be the predicted difference in estimated mean earnings for a white women and a black women?

- In Model 5, Holding other variables constant, what will be the predicted difference in estimated mean earnings for a white man and a black women?

```r
## our code here
earnings <- read.csv("data/lab5_earnings.csv")

## recode
earnings[earnings$age == 9999, "age"] <- NA
earnings <- earnings %>%
  mutate(age =
           case_when(age == 9999 ~ NA,
                     .default = age))

## create dummies for female
earnings <-
  earnings %>%
  mutate(female =
           case_when(sex == "female" ~ 1,
                     .default = 0))

## create dummies for black
earnings <-
  earnings %>%
  mutate(black = case_when(race == "black" ~ 1,
                           .default = 0),
         other = case_when(race == "other" ~ 1,
                           .default = 0))

## fit model 5
lm5 <- lm(earn ~ age + edu + female + black + other + edu*female,
   data = earnings)

summary(lm5)
```

```
##
## Call:
## lm(formula = earn ~ age + edu + female + black + other + edu *
##     female, data = earnings)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.6168  -5.6167   0.2579   5.3951  21.8480
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.97353    1.25357  13.540  < 2e-16 ***
## age          0.15606    0.01923   8.115 1.46e-15 ***
## edu          6.08265    0.14290  42.567  < 2e-16 ***
## female      -1.57109    1.31104  -1.198    0.231
## black       -2.38473    0.55682  -4.283 2.03e-05 ***
## other       -0.94604    1.01653  -0.931    0.352
## edu:female  -3.12819    0.19921 -15.703  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.879 on 973 degrees of freedom
##   (20 observations deleted due to missingness)
```

```
## Multiple R-squared:  0.7983, Adjusted R-squared:  0.797
## F-statistic: 641.8 on 6 and 973 DF,  p-value: < 2.2e-16
```